

APPLICATION
FOR
UNITED STATES LETTERS PATENT

Be it known that we, Sorcha O'Callaghan, a citizen of the Republic of Ireland, residing at 13 Ballyowen Way, Lucan, Co. Dublin, Ireland, Jerome Nolan, a citizen of the Republic of Ireland, residing at 8 Sandyford Downs, Sandyford, Dublin 18, Ireland, Paul O'Keeffe, a citizen of the Republic of Ireland, residing at 54 Bow Bridge Place, Kilmainham, Dublin 8, Ireland, David Nolan, a citizen of the Republic of Ireland, residing at 29 Priory Way, St Raphaels Manor, Celebridge, Co. Kildare, Ireland and Kevin Jennings, a citizen of the Republic of Ireland, residing at 4 Woodstown Way, Knocklyon, Dublin 6, Ireland have invented new and useful improvements in:

NETWORK UNIT WITH ADDRESS CACHE FOR FREQUENTLY OCCURRING
NETWORK CONVERSATIONS

of which the following is a specification:

NIXON & VANDERHYE P.C.
ATTORNEYS AT LAW
1100 NORTH GLEBE ROAD
8TH FLOOR
ARLINGTON, VIRGINIA 22201-4714
(703) 816-4000
Facsimile (703) 816-4100

NETWORK UNIT WITH ADDRESS CACHE FOR FREQUENTLY OCCURRING NETWORK CONVERSATIONS

Field of the Invention

5 This invention relates to network units, that is to say multi-port units which can receive and transmit addressed data packets, such as, for example, Ethernet data packets and which need to perform address look-ups so as to relate address data, which maybe for example media access control address data or network address data or both in order to determine by means of
10 a database relating address data to forwarding data (such as a numerical identification of a respective port or ports) which port or ports are required to send a data packet previously received by the unit. Examples of such network units are "switches", "bridges" and "routers". The invention more particularly relates to making a look-up process more efficient, at least in respect of addresses which appear in more frequently occurring conversations (i.e. message
15 traffic between specific representative by sources and destinations).

Background to the Invention

Various architectures and associated software are available for network units which are
20 required to perform address look-ups to determine the physical destination or destinations of a packet received by a network unit of the kind to which the invention relates. A common feature of such units is a look-up table or "forwarding database" which, despite having a variety of forms and different manners of organization dependant on circumstances, in essence relates address data in a packet to forwarding data. Forwarding databases are
25 generally more complicated than the simple description indicates. For example, they may cope with different layer addresses, have a variety of means of restricting unnecessary look-ups, and employ "aging" to remove entries which have not been used for a particular time. Furthermore, there is a variety of ways of organizing the look-ups in such a database. Owing to the considerable lengths of addresses and the high cost both financially and in temporal
30 terms, of database searching, various schemes have been employed to reduce the memory space required and to accelerate the search process. One method of reducing the space required is to employ "hashing" of address data. The search process may be organised on a

binary tree basis but other forms of searching, such as “trie” searching, are now commonplace.

More complex databases may require more than the destination address of a packet for a successful look-up. For example when the network unit is used in conjunction with a ‘farm’ of servers and all the servers in the farm have the same ‘virtual’ IP (internet protocol) address, the unit, in this example a router, may store traffic among the servers according to the IP source addresses of the traffic. A look-up table for this purpose can be based on a trie search using a key identifying both the network destination address (i.e. the servers’ virtual IP address) and the IP source address, and if appropriate the identification of the virtual LAN of the source.

Such a key may be used to search a trie data structure until a leaf node is found, i.e the look-up is performed by traversing the trie structure until an associated data block containing the ‘forwarding data’ is reached or the search terminates without success. Each stage in the trie uses a few more bits from the key. This means that many addresses with a common prefix can refer to the same target address.

By way of further example, load balancing is a method of sharing the load between a multiplicity of servers for higher performance and for the protection of redundancy. All the servers in a server “farm” may have the same virtual internet protocol address. The router, such as the network unit 10 shown in Figure 1, shares traffic destined for this virtual internet protocol address between the servers based on some selective algorithm. One typical algorithm employs the internet protocol source address of a packet to determine which server will be selected for handling such a packet.

Currently therefore in server load balancing (and also IP multicasting) the entire internet protocol destination address and internet protocol source address need to be stored in a trie look-up table. This means that the search of a very long key may be required. Very typically, in a trie search the key is read in slices, and typically ten “slices” are required. This consumes substantial time and it is desirable to improve the performance of the switch by rendering more common reads are necessary.

It is known, for a variety of purposes, to provide a "cache" memory, for example constituted by a content addressable memory (CAM). Such a memory can be accessed by direct application of address data, for example as described in European patent application EP-0594196-A1, wherein the content addressable memory is employed in conjunction with hashing of address data to provide a search key and particularly to deal with cases where more than a particular number of addresses would hash to the same key.

Content addressable memories, for example as described in US patent 4587610, require only one read operation to compare all stored addresses with an incoming address. However, this speed of operation is obtained at the cost of great complexity and cost. The present invention will employ a content addressable memory only for a small number of address pairs.

It is known from Wakeman et al, US Patent 5740175 issued 14 April 1998, to provide a network switch which includes a RAM forwarding database containing the address to port mappings for all the workstations or other devices connected to the switch's plurality of ports and at least one CAM-cache connected to a respective switch port so that when it is desired for the switch to forward a packet, the destination address is extracted and if the correct mapping is contained in the CAM-cache the packet is immediately forwarded to the destination port without accessing the much larger and slower forwarding database. Wakeman et al provide for updating of the CAM-cache when it is full by inserting a new mapping for either the least recently used mapping or the least frequently used mapping in the database.

A known technique in association with network units is known as "RMON" (remote monitoring). This in essence comprises remotely controlled "snooping" on packets which pass through the switch in order to obtain information from the header portions of the packets. Remote monitoring is employed for a variety of purposes, particularly the analysis of traffic flows in the system to which the network unit is connected. It is described in a variety of publications, such as for example in GB-2317542, GB-2316589, 2338147 and so on. A known feature of remote monitoring is the identification of each "conversation" (identified by a pair of addresses) which is handled by the unit at any particular time.

Summary of the Invention

The present invention has the object of avoiding or at least reducing the number of look-ups in a forwarding database that may be necessary by maintaining in a cache entries identifying frequently used 'conversations' occurring in packets handled by the unit together with the relevant forwarding data, or at least a means of accessing the relevant forwarding data for such packets. The cache memory may be constituted by a content addressable memory which can rapidly provide access to the relevant forwarding data.

In a preferred form of the invention a look-up process comprises accessing the cache to determine whether a conversation, defined by an address pair, is the subject of an entry in the cache and resorting to a larger database if the conversation is not the subject of an entry in the cache.

The cache is preferably established and maintained by recourse to the table or list of conversations, preferably established by RMON techniques. In particular, as part of the look-up process, if a measure of traffic flow (such as a packet count) relevant to a first conversation represented by a packet is greater than that for a second conversation which is the subject of an entry in the cache, the first conversation (if not in the cache) may be put in the cache in place of the second. The cache may be continually updated in this manner.

Further features of the invention will be apparent from a description of a more detailed example, with reference to the drawings.

Brief Description of the Drawings

Figure 1 is a schematic diagram of a network unit which would typically host the present invention.

Figure 2 illustrates a trie search.

Figure 3 illustrates a conversation table.

Figure 4 illustrates a search process according to the invention.

Figure 5 illustrates the interrelationship between the main functional parts participating in a search process according to the invention.

Figure 6 is a flow diagram illustrating in simplified form a search process according to the invention.

Figure 7 is a flow diagram illustrating the search process more particularly in relation to the access to and control of the cache memory.

Description of a Preferred Example

Figure 1 of the drawings illustrates in a schematic manner a network unit 10 in accordance with the invention. The unit is intended to represent one example of a wide variety of units which can host the present invention.

Merely by way of example, the unit shown in figure 1 has a multiplicity of ports, represented in the example by four ports 11, 12, 13 and 14. These ports are adapted for connection to an external network by any suitable medium, for example twisted pair, coaxial line, fibre optic line or wireless link according to circumstances. The ports 11 to 14 contain as is usual physical layer control and media access control, for example in accordance with the relevant transmission standard.

The unit includes a bus system which is shown as the bus 15 for the conveyance of control data, packet data and such like around the various parts of the unit. Although the ports 11 to 14 may have temporary buffers associated with them, in accordance with known practice, storage of packets that have been received and before they are transmitted is effected in a central packet memory 16. In order to determine where a given packet should go, a look-up engine 18 is provided. In essence this includes a database (e.g. as shown in Figure 2) which contains entries relating address data to forwarding data (a number or other data identifying a

port) so that for any given destination (which may be single or multiple) the accessing of the look-up table from the relevant data will yield the forwarding data which is used by a forwarding engine, in this example represented by the CPU 17, which will cause extraction of the relevant packet from memory 16 and the forwarding of the packet to the relevant port or ports.

As thus so far described the unit is commonplace and various different architectures are known and available commercially. The remaining blocks in Figure 1 will be described later.

As is more particularly described later, for a variety of purposes, such as load balancing in a server farm, it is desirable or necessary to perform look-ups using both source and destination data of a packet, and for this purpose a trie search facility may be provided.

Figure 2 illustrates a known manner of organising a search table according to a trie look-up scheme. The look-up table is organised in blocks to two types. One type, exemplified by blocks 53, 54 and 55, contain pointers which are either of a trie type, such as pointer 56, that point to another block of trie type or, as exemplified by pointer 60, are of an 'associated data' type which point to the second type of block, the AD block exemplified by blocks 59 and 61. The AD or associated data blocks contain the forwarding information for the relevant address key.

The first block, 53, in the trie search is accessed by the first set of bits from the trie key. These bits identify a pointer within the block 53. In the case of a pointer such as pointer 56, which points to block 54, the pointer within block 54 will be accessed by the second set of bits from the trie key and so on until an associated data block, such as block 59, is reached.

It may be in some cases that a group of addresses share the same route; then the first part of the trie key will immediately yield, as in the case of pointer 60, a pointer to an associated data block 61 and it is then unnecessary to traverse the whole key in order to obtain the forwarding information.

Figure 2 shows two keys 51 and 52 which are long keys composed of the internet protocol destination address and the internet protocol source address. They may also contain VLAN information.

5 For further information on the organisation of trie searches, reference may be made to European patent application EP-0551243-A2, US patent 6041053 (Douceur et al) and international patent application publication number WO96/00945.

10 As indicated in the foregoing, currently an entire address pair, the internet protocol destination address and internet protocol source address needs to be stored in a trie look-up table preferably with the number of the source VLAN. This means a search of the entire key may be needed. Typically, the key is read in slices of bits, for example an initial segment of eleven bits, subsequent segments of seven bits and a final segment of four bits, requiring up to ten reads from the trie search table.

15 A further known feature in switches of this kind is a remote monitoring facility, shown in Figure 1 as "RMON" engine 19. In this example the remote monitoring system is coupled to the internal bus 15, as described in British patent GB-2316589 and elsewhere.

20 Remote monitoring is known to provide a "conversation table" which is essentially a table indexed according to address pairs together with an indication of the number of times that address pair has occurred.

25 There is a variety of ways in which the remote monitoring can be performed. It may be done randomly, or at regular intervals. The measure of traffic flow for any given address pair maybe in terms of bytes or packets.

30 The RMON engine 19 monitors traffic that is bridged and/or routed by the unit. For each packet (or each of a succession of sample packets at regular or pseudorandom intervals) entering the switch it checks whether an entry for a conversation pair has been entered in a table in static random access memory. If it has not been so entered, the RMON engine creates a new entry. If an entry does exist it updates the statistics for the conversation.

The address pairs for the most active conversations as determined by the RMON engine are put into a cache memory, namely the CAM (content addressable memory) 21. Along with each entry will be a pointer to the associated data block for this conversation. The access to the CAM 21 and its updating is controlled by a cache controller 20, preferably constituted by a state machine which will be described with reference to Figure 7. For each packet entering the switch the relevant destination and source address data will be entered into the cache if this conversation has more frequently occurred than the least frequent in the cache and if it has not been entered before. Entries may also be aged out if they are inactive for a specified time. Each time the associated data block is added or removed or a branch of layer 3 trie search is modified, the AD block pointer in the cache will be updated accordingly.

When a layer 3 packet is about to be looked up, its destination address and source address will be compared with the entries in the cache. If the entry is found the pointer will be used to go straight to the AD block. This saves many reads and enables routing of these packets at the wire rate.

A variation on this method is to have the associated data block located in the cache also. In this case each time the information and the associated data block is modified the corresponding entry in the cache needs to be updated.

Figure 3 illustrates a known form of conversation table 21 which can be constructed using a known RMON engine 19. The monitoring system is, for example, coupled to the main data bus of the switch so that it can snoop on packets (or a regular or random selection of packets) in order to obtain, at each sample, the IP destination and source addresses of the sampled packet. The conversation table also needs a measure of the traffic; this may be represented by a count of packets or may be represented by a byte count according to preference. The conversation table thus has entries schematically represented in Figure 3, each of which identifies an IP source and destination pair and contains an indication of the packet count flowing between that pair. The source addresses are SA, SB etc and the destination addresses are DA, DB etc. The values T1, T2 etc represent the volume of traffic containing the respective pair of addresses.

Figure 4 illustrates in a generally schematic form a search process according to the invention.

Block 70 represents the known process by which packets are received into the switch, temporarily stored in packet memory and so on. Block 71 represents the operation of the RMON engine, which maintains a table for the most frequent conversations, as obtained by examination of the packet headers as more fully described later. Line 72 represents the updating of the RMON engine as each new packet is received by the switch. Arrow 73 represents a look-up request performed in respect of a packet while it is stored in the switch in order to obtain the forwarding data for the particular packet.

Block 74 represents the search process performed according to the invention. The pair of addresses (IPSA and IPDA) are compared with the entries in the cache, which is normally a fixed size. The cache is populated with addresses as they are seen (Figure 7). If the address pair is not in the cache, the 'ordinary' look-up process will be initiated, as denoted by arrow 75. That process will normally yield the forwarding data, so that the packet is forwarded (arrow 76) to one or other of the ports.

Owing to the nature of the RMON engine, the entries in the cache are not necessarily the most frequently occurring conversations at any given time, though statistically they will constitute most of the frequent conversations.

The search process is summarised in Figure 6. From a start 80 the addresses obtained from a packet, stage 81, are read. The cache is accessed, stage 22. The cache will be populated with addresses as they are seen. If the address pair, the IP destination address and IP source address are in the cache (stage 83), then the associated data block (stored in the cache along with the address pair) is read (stage 85), the forwarding data obtained (stage 86) and the process terminated. If the address pair is not available from the cache, a look-up will be performed by means of the trie search facility (stage 84) as previously described with reference to Figure 2.

Figure 5 shows the interrelationship between the principal functional parts of the search process, involving the cache 21, the RMON engine 19, the cache controller 20 and the look-up engine. The RMON engine provides packet counts and address pairs. The cache controller 20 maintains a threshold and the number of entries filled in the cache, in registers 171 and 172. The cache controller has recourse to the cache to determine whether the address pair is matched against data held in the cache and the cache will signal the cache controller to signify a match. The cache controller will also push an entry on to the cache under certain conditions to be described later.

Figure 7 illustrates more specifically the operation of the cache controller. The start of the process shown by Figure 7 is the commencement 101 of the processing of the packet. At stage 102 the controller determines whether the cache is fully populated. Initially it will be assumed that the cache is fully populated so that the next stage 103 is a determination whether the conversation (the particular IP source and destination address pair) is in the cache. If the conversation is in the cache, there is a determination 104 whether the packet count is equal to a value held in threshold register 171 incremented by unity. If it is, then the threshold register is updated (105) with the new packet count.

Then the AD information from the cache entry (or elsewhere) is obtained (106) and a jump is made to the end of the look-up routine.

If the conversation is not in the cache several actions will be required. One (107) is to perform the ordinary look-up (e.g. as shown in Figure 2). At stage 108 the controller determines whether the packet count for the conversation is greater than that in the threshold in register 171. If the packet count for the conversation is not greater than the threshold register there will be no need to enter this conversation in the cache and the process will terminate in readiness for the processing of the next packet. If however the packet count for the conversation is greater than the count in the threshold register then the least frequent conversation in the cache will be replaced (stage 109) by the conversation represented by the new packet, including the AD block when it is found. The threshold register will be updated with the packet count for this conversation.

If however the cache is not fully populated then the new conversation may be entered into the cache. A determination (110) will be made whether the conversation is in the cache already. If it is, then the AD information from the cache is employed (stage 111) and there is a jump to the end of the look-up routine. If the conversation is not in the cache, and the cache is not
5 fully populated, the conversation must be entered in the cache (stage 112). This will be accompanied by the AD block. There needs also to be a determination (stage 113) whether the packet count for the packet is greater than the number in the threshold register. If it is not, then the threshold register will be updated (stage 114) with the packet count.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
220